Butterworth Low-pass Filter Math

by Kenneth A. Kuhn
March 6, 2013

The Butterworth response is known as the maximally flat response because there are no response ripples and the response remains at near unity for as much of the pass-band as possible. Examples are shown in Figure 1.

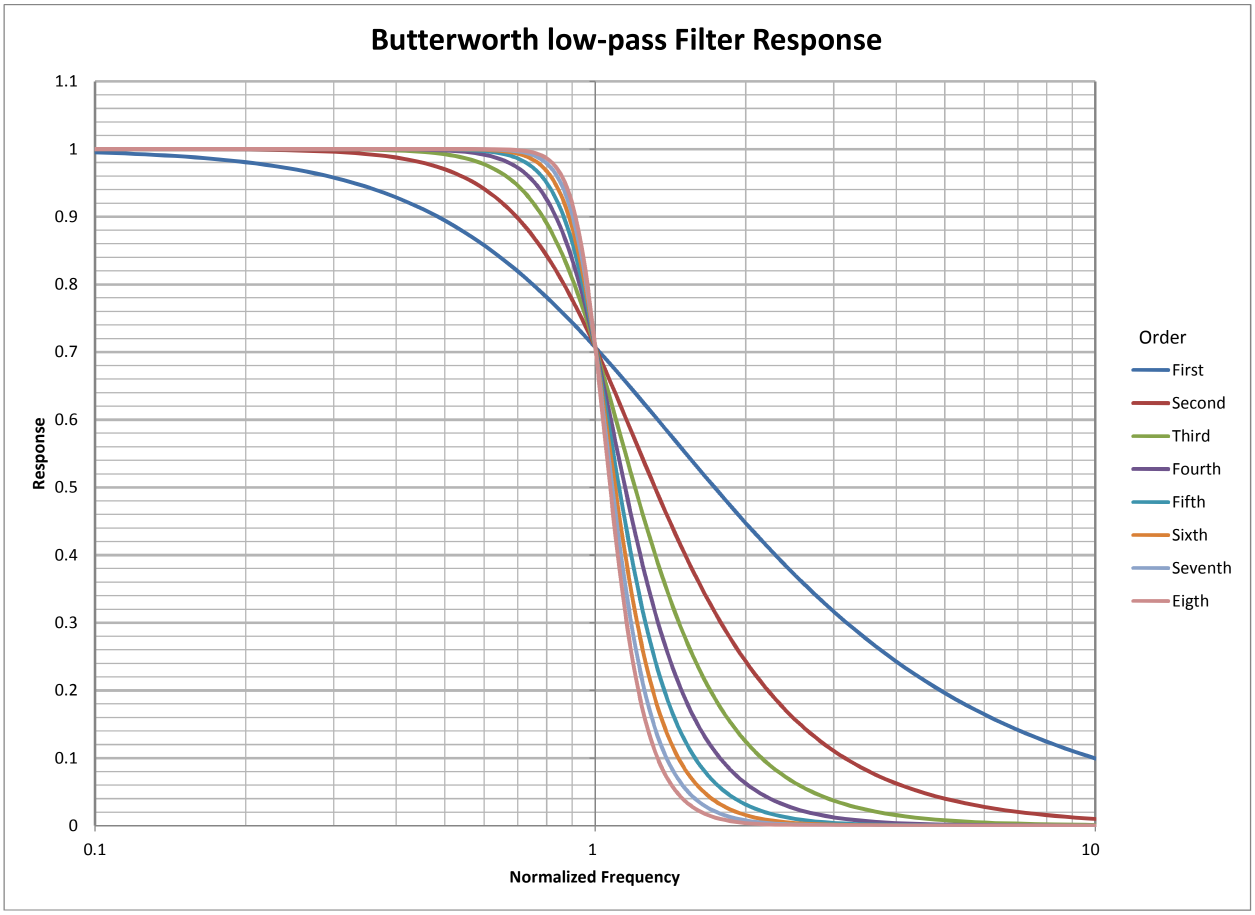

Figure 1: Examples of Butterworth low-pass filter response

The responses shown in Figure 1 are readily calculated using Equation 1 which is a convenient result of the fact that the poles of the filter are located equally on a circle. Ordinarily, calculating the frequency response of high order systems is quite involved mathematically.

$$R = \frac{1}{\sqrt{1 + \left(\frac{F}{F_c}\right)^{2n}}} \qquad (1)$$

R is the transfer function of the filter, $Vo_{(s)}$ / $Vin_{(s)}$
F is the frequency of interest

Butterworth Low-pass Filter Math

F_c is the -3dB cutoff frequency of the filter
n is the order of the filter, 1, 2, 3, etc.

Given a set of filter specifications, F_P, R_P, F_S, R_S as shown in Figure 2 our design job is to determine the minimum order, n, and the optimum cutoff frequency, F_C to meet the specification.

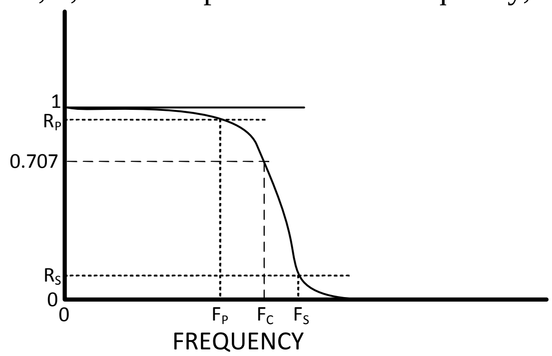

Figure 2: Setup for design of Butterworth low-pass filter

F_P is the highest frequency of interest in the pass-band of the filter.
R_P is the minimum pass-band response factor relative to the DC response we can tolerate.
F_S is the lowest frequency of interest in the stop-band of the filter.
R_S is the maximum stop-band response factor relative to the DC response we can tolerate.
F_C is the -3dB cutoff frequency of the filter.

The first step is to setup a system of two equations with the two unknowns, n and F_C. We insert our filter specifications into the two equations as follows.

$$R_S = \frac{1}{\sqrt{1 + \left(\frac{F_S}{F_c}\right)^{2n}}} \tag{2}$$

$$R_P = \frac{1}{\sqrt{1 + \left(\frac{F_P}{F_c}\right)^{2n}}} \tag{3}$$

We next square both sides of each equation to get rid of the radical.

$$R_S^2 = \frac{1}{1 + \left(\frac{F_S}{F_C}\right)^{2n}} \tag{4}$$

Butterworth Low-pass Filter Math

$$R_P^2 = \frac{1}{1 + \left(\frac{F_P}{F_C}\right)^{2n}} \tag{5}$$

The next step is to rearrange so that the frequency ratios are on the left side.

$$\left(\frac{F_S}{F_C}\right)^{2n} = \frac{1}{R_S^2} - 1 \tag{6}$$

$$\left(\frac{F_P}{F_C}\right)^{2n} = \frac{1}{R_P^2} - 1 \tag{7}$$

Observe that if we take the ratio of the two equations F_C vanishes and the only unknown is *n*.

$$\left(\frac{F_S}{F_P}\right)^{2n} = \frac{\frac{1}{R_S^2} - 1}{\frac{1}{R_P^2} - 1} \tag{8}$$

We take the natural logarithm of both sides.

$$2n \ln\left(\frac{F_S}{F_P}\right) = \ln\left(\frac{\frac{1}{R_S^2} - 1}{\frac{1}{R_P^2} - 1}\right) \tag{9}$$

Now we can solve for *n*.

$$n = \frac{\ln\left(\frac{\frac{1}{R_S^2} - 1}{\frac{1}{R_P^2} - 1}\right)}{2 \ln\left(\frac{F_S}{F_P}\right)} \tag{10}$$

It is highly improbable that *n* will be an integer. Most likely *n* will be something like 5.67. It is only practical to build integer order filters. So do we round *n* up or down to the nearest integer? The general rule is to round up to the next higher integer. However, we must realize that perhaps the filter specifications probably contain some amount of arbitrary judgment calls that often result in a large filter order being required. We should first review the specifications, particularly if *n* is greater than about six, to see if some values can be relaxed so that a lower order and thus a more practical filter results. So although technically you should round up to six even if *n* is 5.12 you have an opposing force to see if you can make the filter as low order as will truly do the required job. It is a balancing act and computer simulation can be of great help in making the decision.

The end result, regardless of the process, is that you will at this point have an integer value for *n*. The next step is to calculate the required F_C by substituting *n* into either Equation 2 or 3. If you could use the exact value of *n* there would be a unique solution to F_C independent of which Equation (2 or 3) you used. In that special case the response curve would exactly go through the R_P and R_S points as shown in Figure 2. With a modified integer value for *n* then there will be two solutions for F_C depending on which equation is used. In one solution we obtain the lowest possible cutoff frequency, F_{CL}, and the response curve will exactly go through the R_P point but fall below the R_S level at a lower frequency than F_S – the filter is better than the stop-band specification as shown in Figure 4. In the other solution we obtain the highest possible cutoff frequency, F_{CH}, and the response curve will exactly go through the R_S point but be higher than the R_P level in the pass-band – the filter is better than the pass-band specification as shown in Figure 5.

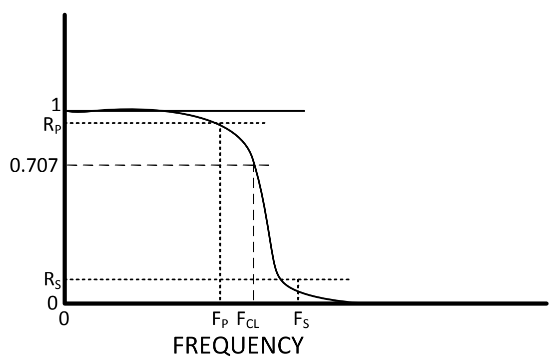

Figure 4: Filter exactly meets the pass-band specification
and is better than the stop-band specification

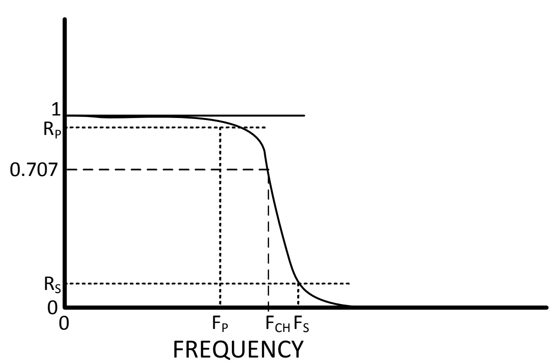

Figure 5: Filter exactly meets the stop-band specification
and is better than the pass-band specification

Which of the two possible solutions should we use? The answer is to compute both solutions and use the average cutoff frequency, F_{CA}. The resulting filter will be better than the pass-band specification and also will be better than the stop-band specification as illustrated in Figure 6. This result gives us the best of both worlds.

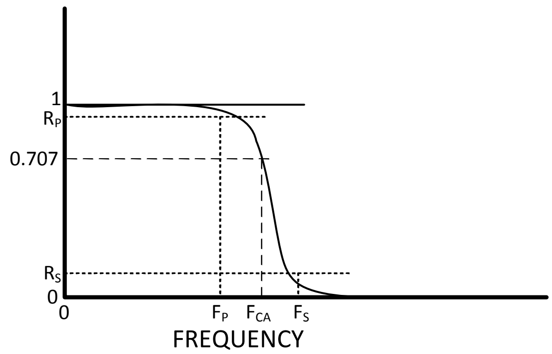

Figure 6: Using the average cutoff frequency from the two solutions to exceed both the pass-band and stop-band specifications

We first solve the following equation for the lowest possible cutoff frequency, F_{CL}.

$$R_P = \frac{1}{\sqrt{1 + \left(\frac{F_P}{F_{CL}}\right)^{2n}}} \tag{11}$$

In a few steps that leads to:

$$F_{CL} = \frac{F_P}{e^{\left(\frac{\ln\left(\frac{1}{R_P^2} - 1\right)}{2n}\right)}} \tag{12}$$

We next solve the following equation for the highest possible cutoff frequency, F_{CH}.

$$R_S = \frac{1}{\sqrt{1 + \left(\frac{F_S}{F_{CH}}\right)^{2n}}} \tag{13}$$

In a few steps that leads to:

$$F_{CH} = \frac{F_S}{e^{\left(\frac{\ln\left(\frac{1}{R_S^2}-1\right)}{2n}\right)}} \tag{14}$$

We will use the average cutoff frequency, F_{CA}.

$$F_{CA} = \frac{F_{CL} + F_{CH}}{2} \tag{15}$$

The preceding assumes that the rounded *n* is higher than the result calculated from Equation 10. If *n* was actually less – perhaps driven by practicality, then the F_{CL} and F_{CH} calculations would numerically swap but the average of the two is still the best value to use.

We calculate the system poles using Equation 16 by letting *k* go from 0 to *n*-1 although by symmetry we only have to compute to *n*/2 (and rounding any 0.5 result up to the next integer). Note that $\omega_C = 2\pi F_{CA}$.

$$pole_k = \omega_C \left\{ \cos\left[\frac{\pi}{2}\left(1 + \left(\frac{2k+1}{n}\right)\right)\right] + j\sin\left[\frac{\pi}{2}\left(1 + \left(\frac{2k+1}{n}\right)\right)\right]\right\} \tag{16}$$

These poles lie on a circle in the left-half s-plane as shown in Figure 7. Note that the poles are evenly spaced and that the radius of the circle is ω_C.

Butterworth Low-pass Filter Math

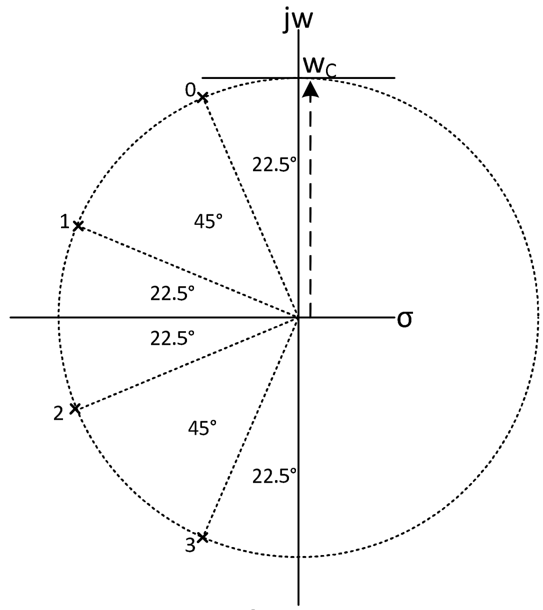

Figure 7: Pole Locations for a 4^{th} order Butterworth Low-pass Filter

See the following web link for more information.

http://en.wikipedia.org/wiki/Butterworth_filter